What is claimed is:

1. A process for use in a database system, comprising:

storing data according to a first user-defined data type in a table;

associating at least a first compression routine with the first user-defined data type; and

using the first compression routine to compress the data according to the first user-defined data type.

2. The process of claim 1, further comprising using a second compression routine to compress the data to improve compression efficiency.

3. The process of claim 2, wherein using the first and second compression routines comprises using user-defined data type methods.

4. The process of claim 3, wherein using the user-defined data type methods comprises using methods built in with the first user-defined data type.

5. The process of claim 1, wherein using the first compression routine comprises using a first compression method built in with the first user-defined data type.

6. The process of claim 5, further comprising providing a user-defined method executable to invoke the first compression method.

7. The process of claim 6, further comprising invoking the user-defined method to invoke a second compression method built in with the first user-defined data type.

8. The process of claim 7, wherein invoking the user-defined method comprises invoking the user-defined method to alter compression efficiency.

9. The process of claim 1, further comprising providing a second user-defined data type built upon the first user-defined data type.

10. The process of claim 9, further comprising storing a first type of data using the first user-defined data type and storing a second type of data using the second user-defined data type.

11. The process of claim 10, further comprising using a second compression routine to compress the second type of data.

12. The process of claim 9, further comprising inheriting at least a data structure and at least a built-in method from the first user-defined data type into the second user-defined data type.

13. An article comprising at least one storage medium containing instructions that when executed cause a system to:

store data according to a first user-defined data type; and

associate a first compression routine with the first user-defined data type for compressing the data.

14. The article of claim 13, wherein the instructions when executed cause the system to associate a second compression routine with the first user-defined data type, the first and second compression routines providing different compression algorithms.

15. The article of claim 14, wherein the instructions when executed cause the system to provide the first compression routine as a method built in with the first user-defined data type.

16. The article of claim 15, wherein the instructions when executed cause the system to provide the second compression routine as a method built in with the first user-defined data type.

17. The article of claim 13, wherein the instructions when executed cause the system to associated a first data structure with the first user-defined data type, the first data structure to indicate a type of compression applied on a data object.

18. The article of claim 17, wherein the instructions when executed cause the system to associate a second data structure with the first user-defined data type, the second data structure to indicate a percentage amount of compression of the data object.

19. The article of claim 18, wherein the instructions when executed cause the system to access the first and second data structures of the data object when accessing the data object.

20. The article of claim 19, wherein the instructions when executed cause the system to store the data object in a relational table.

21. The article of claim 19, wherein the instructions when executed cause the system to store the data object in a relational table distributed across multiple access modules.

22. The article of claim 20, wherein the instructions when executed cause the system to provide a second user-defined data type built upon the first user-defined data type.

23. The article of claim 13, wherein the instructions when executed cause the system to provide a second user-defined data type built upon the first user-defined data type.

24. The article of claim 23, wherein the instructions when executed cause the system to inherit the first compression routine from the first user-defined data type into the second user-defined data type.

25. The article of claim 24, wherein the instructions when executed cause the system to:

associate a second compression routine with the first user-defined data type; and

inherit the second compression routine from the first user-defined data type into the second user-defined data type.

26. The article of claim 25, wherein the instructions when executed cause the system to:

store a first type of data using the first user-defined data type; and

store a second type of data using the second user-defined data type.

27. A database system, comprising:

a storage system to store at least a table;

a plurality of compression routines to apply respective different compression algorithms; and

a controller adapted to invoke one of plurality of compression routines to compress data stored in the table.

28. The database system of claim 27, wherein the table includes a relational table and the data is stored in a first attribute of the relational table.

29. The database system of claim 28, wherein the first attribute is according to a first user-defined data type.

30. The database system of claim 29, wherein the plurality of compression routines are methods built in with the first user-defined data type.

31. The database system of claim 30, the storage system to store a second table having a second attribute according to a second user-defined data type built upon the first user-defined data type.

32. The database system of claim 27, wherein the controller is adapted to invoke another one of the compression routines to alter compression of the data.

33. The database system of claim 32, wherein the controller is adapted to invoke another one of the compression routines in response to a Structured Query Language UPDATE statement.

34. The database system of claim 33, wherein the controller comprises a user-defined method.

35. The database system of claim 34, wherein the plurality of compression routines comprise methods built in with the first user-defined data type,

the user-defined method executable to invoke the methods built in with the first user-defined data type.

36. The database system of claim 27, further comprising a plurality of access modules adapted to manage access to respective portions of the storage system.

37. The database system of claim 36, wherein the table is distributed across multiple access modules.